**96 Overheads**

Part 6e: Cascading Style Sheets (CSS)

More about this course about Internet application protocols can be found at URL:
Last update: 04-09-19 19.39

---

**Why is CSS good?**

CSS separates formatting from logical content, which has many advantages,

1. formatting codes need not be repeated,
2. it is easier to produce text if it is separated from formatting mark-up,
3. the risk of errors is probably smaller,
4. it is easier to format the same content in multiple formats,
5. it is easier to modify the formatting for many pages,
6. tasks can be better separated between content producer and format designer.

---

**Rendering:**

Here is one word, which is boldfaced and with a smaller font.

**Old notation:**

Here is one `<B><FONT SIZE="2">word</FONT>` which is boldfaced and with a smaller font.

**New notation:**

```html
<HEAD>
<STYLE type="text/css">SPAN.xyz { font-size:10pt; font-style: bold }</STYLE>
</HEAD><BODY>
Here is one <SPAN class="xyz">word</SPAN> which is boldfaced and with a smaller font.
</BODY>
```

**Note:** { } surrounds a group of properties for a certain style.

---

**Relative and absolute font sizes**

Before we got style sheets, HTML only allowed relative font sizes. A user could specify `<font size="n">` where the following values of “n” were allowed:

<table>
<thead>
<tr>
<th>Non-signed value</th>
<th>Signed value</th>
<th>Example of possible rendering, for a Windows user with 12 pt as default font size using the Arial font (50 % larger here)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-2</td>
<td>The quick brown fox</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>The quick brown fox</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>The quick brown fox</td>
</tr>
<tr>
<td>4</td>
<td>+1</td>
<td>The quick brown fox</td>
</tr>
<tr>
<td>5</td>
<td>+2</td>
<td>The quick brown fox</td>
</tr>
<tr>
<td>6</td>
<td>+3</td>
<td>The quick brown fox</td>
</tr>
<tr>
<td>7</td>
<td>+4</td>
<td>The quick brown fox</td>
</tr>
</tbody>
</table>
With CSS, absolute fonts became possible

With CSS, absolute fonts became possible, for example

```html
<style type="text/css">
    .arial8 { font-family: Arial; font-size: 8pt }
</style>
```

Absolute font size Windows rendering (96 pixels/inch) Macintosh rendering (72 pixels/inch)

<table>
<thead>
<tr>
<th>Font size</th>
<th>Windows</th>
<th>Macintosh</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 pt</td>
<td>Arial 8 pt</td>
<td>Arial 8 pt</td>
</tr>
<tr>
<td>10 pt</td>
<td>Arial 10 pt</td>
<td>Arial 10 pt</td>
</tr>
<tr>
<td>12 pt</td>
<td>Arial 12 pt</td>
<td>Arial 12 pt</td>
</tr>
</tbody>
</table>

Note: The pictures have been increased 50% (with corresponding reduced resolution) to become more readable on an overhead screen.

Command links should not change colour

Below, a command bar is produced using a HTML table (saves download time compared to using graphical buttons). But the links in the command bar should not change colour when they have been visited, like the links in the text below the command bar.

Visited and unvisited links have the same colour:

<table>
<thead>
<tr>
<th>Hem</th>
<th>Personligt</th>
<th>Logga ut</th>
<th>WebGroups</th>
<th>Skrea hrov</th>
<th>Hjelp</th>
</tr>
</thead>
</table>

Visited and unvisited links have different colour:

<table>
<thead>
<tr>
<th>Hem</th>
<th>Personligt</th>
<th>Logga ut</th>
<th>WebGroups</th>
<th>Skrea hrov</th>
<th>Hjelp</th>
</tr>
</thead>
</table>

Archives of this list are available by anonymous ftp from FTP://SEGATE.SUNET.SE. You can also browse the archives by http from HTTP://segate.sunet.se/archives/mailnews-1.html. The FTP archives are better if you want to download all messages, the HTTP archives are better if you want to browse and find a particular message only.

How to stop links changing colour when visited?

Old method: only works for all links in the whole document:

```html
<BODY BGCOLOR="#FFFFFF" LINK="#003399" VLINK="#003399" ALINK="#003399">

Using style sheets (the HTML below is abbreviated):

```html
<HTML><HEAD>
<TITLE>CSS and link colour</TITLE>
<STYLE type="text/css">
    A.allblue:visited { color: #003399 }
    A.allblue:link { color: #003399 }
</STYLE></HEAD><BODY BGCOLOR="#FFFFFF"><FONT SIZE=5>

Visisted and unvisited links have the same colour:

```

```html
<TABLE BORDER="1" CELLPADDING="4" CELLSPACING="1" WIDTH="500">
  <TR><TD BGCOLOR="#FFCC00">
      <A class="allblue" HREF="hem.html" target="subwindow">
          <FONT FACE="Geneva, Helvetica" SIZE=3>
          <B>Hem</B></A></TD></TR>
  <TR><TD BGCOLOR="#FFCC00">
      <A class="allblue" HREF="hem.html" target="subwindow">
          <FONT FACE="Geneva, Helvetica" SIZE=3>
          <B>Hem</B></A></TD></TR>
</TABLE>
```

This link will not change colour

Using CSS to avoid too long rows

For good readability, text rows should contain about 40-60 characters per line. But if the user has set his browser to a wide window (which the user needs when viewing other web pages) ordinary text will be too wide.

Archives of this list are available by anonymous ftp from FTP://SEGATE.SUNET.SE. You can also browse the archives by http from HTTP://segate.sunet.se/archives/mailnews-1.html. The FTP archives are better if you want to download all messages, the HTTP archives are better if you want to browse and find a particular message only.

This shows Netscape rendering of text shows the same text in three ways:

1. Ordinary plain text with a wide browser window
2. Using a table with `<TD WIDTH=400>`
3. Using style sheets with `<P class=`
Using CSS to avoid too long rows

\[
\text{HTML}\langle \text{HEAD} \\
\text{TITLE}>\text{CSS and link colour</TITLE> \\
\text{STYLE type="text/css"> \\
P.narrow { width: 400 } \\
</STYLE> \\
</HEAD><BODY BGCOLOR="#FFFFFF">
\]

Table method

\[
\text{TABLE BORDER="0" CELLPADDING="8" CELLSPACING="0" WIDTH="400"}\langle TR\langle TD> \\
\text{Iamque fretum Minyae Pagasaea puppe secabant, perpetuaque trahens inopem sub nocte senectam.} \\
\langle/TD\langle/TR\langle/TABLE>
\]

CSS method

\[
\text{<P class="narrow"> \\
Iamque fretum Minyae Pagasaea puppe secabant, perpetuaque trahens inopem sub nocte senectam.</p>
\]

Using style sheets for absolute positioning

\[
\text{HTML}\langle \text{HEAD} \\
\text{TITLE}>\text{CSS position command</TITLE> \\
\text{STYLE type="text/css"> \\
\text{div.dsvtext} \{ position: absolute; left: 135px; top: 10px; right:auto; bottom: auto; width:150 \} \\
\text{div.backimg} \{ position: absolute; left: 300px; top: 60 px; right:auto; bottom: auto \} \\
\text{div.saveimg} \{ position: absolute; left: 40px; top: 130px; right:auto; bottom: auto \} \\
\text{div.flowerimg} \{ position: absolute; left: 20px; top: 130 px; width:100 px; bottom: auto \} \\
\text{div.saveimg} \{ position: absolute; left: 150px; top: 150px; right:auto; bottom: auto \} \\
</STYLE> \\
</HEAD><BODY BGCOLOR="#FFFFFF">
\]

Using style sheets for absolute positioning

(may not work with all browsers)

\[
\text{Department of Computer and Systems Sciences}
\]

\[
\text{Save on Disk}
\]

\[
\text{A flower to CSS}
\]

\[
\text{<DIV class="backimg"><IMG SRC="Arrow.GIF" WIDTH="45" HEIGHT="31" ALIGN="BOTTOM" BORDER="0" ALT="&lt;="></DIV> \\
\text{<DIV class="saveimg"><IMG SRC="disk.gif" WIDTH="32" HEIGHT="29" ALIGN="BOTTOM" BORDER="0" ALT="Disk"></DIV> \\
\text{<DIV class="flowerimg"><IMG SRC="flower1.gif" WIDTH="28" HEIGHT="23" ALIGN="BOTTOM" BORDER="0" ALT="Flower"></DIV> \\
\text{<DIV class="savetext"><FONT SIZE=5>Save on Disk</DIV> \\
\text{<DIV class="flowerimg"><IMG SRC="flower1.gif" WIDTH="28" HEIGHT="23" ALIGN="BOTTOM" BORDER="0" ALT="Flower"></DIV> \\
\text{<DIV class="savetext"><FONT SIZE=5>A flower to CSS</DIV> \\
\langle/BODY></HTML>}
\]
Web page with absolute positioning: Site map

Web4Health Site Map

Advantage: Same background and formatting can easily be used to produce this web page in multiple languages (English, Swedish, etc.)

Extract of CSS used in the site-map example:

```css
.text {
  font-family: verdana, geneva, helvetica, arial, non-serif;
  font-size: 11px
}
#categories { position: fixed; left: 241px; top: 123px
}
```

Extract of HTML used in the site-map example:

```html
<div class="text" id="categories">
  <a href="/en/answers/project-all-menus.htm">Categories</a>
</div>
```

Note: “class” is suitable for formatting which is used many times, “id” for formatting used only once.

Using tables for exact positioning

```html
<table border=0 cellpadding=0 cellspacing=0>
<tr>
  <td rowspan=2 bgcolor="#0066FF">
    <center>
      <font size=7 color="#FFFFFF"><b>*:96</b></font>
    </center>
  </td>
  <td width=73 height=100 bgcolor="#0066FF">
    <img src="exam-98-05a.gif" width=73 height=100>
  </td>
  <td width=254 height=100 bgcolor="#FFFFCC">
    <font size=5><b>Internet application protocols and standards</b></font>
  </td>
</tr>
<tr>
  <td height=14 colspan=2 bgcolor="#3366FF"><br></td>
</tr>
</table>
```

Selectors in CSS

```css
.big { font-size: 24px }

<p class=big>, <div class=big>
  p.big { font-size: 24px }
  <div>
    div.big { font-size: 24px }
  </div>
  p#big { font-size: 24px }
  <div id=big>
    a.dynamic:hover { color: red ; font-weight: bold}
    a.dynamic:link { color: blue ; font-weight: normal}
    a.dynamic:visited { color: red ; font-weight: normal}
  </div>
  h1 em {color: red }
</div>
```

There are many more selector variants in the CSS recommendation.
**Cascading in Cascading Style Sheets**

Style sheets may be specified by

(a) Author
(b) Reader, specially for this document
(c) Defaults in the browser, set at delivery or modified by the user

How, then, should conflicting style sheet information be combined?

---

**Measurements and inheritance:**

<table>
<thead>
<tr>
<th>cm, mm, in</th>
<th>h1 {padding-top: 5mm }</th>
</tr>
</thead>
<tbody>
<tr>
<td>pixels¹</td>
<td>h1 { font-size: 12px }</td>
</tr>
<tr>
<td>points = inch/72</td>
<td>p { word-spacing: 20pt }</td>
</tr>
<tr>
<td>relative to inherited size</td>
<td>h1 {margin: 0,5em }</td>
</tr>
<tr>
<td>ex</td>
<td>h1 {margin-left: 0,5ex }</td>
</tr>
<tr>
<td>%</td>
<td>h1 {line-height: 120% }</td>
</tr>
</tbody>
</table>

1 Converted when printed on paper based on 96 pixels/inch.

---

**Cascading order (section 6.3.1 in the CSS recommendation)**

To find the value for an element/property combination, user agents must apply the following algorithm:

1. Find all declarations that apply to the element/property in question. Declarations apply if the associated selector matches [p. 43] the element in question. If no declarations apply, terminate the algorithm.

2. Sort the declarations by explicit weight: declarations marked 'important' carry more weight than unmarked (normal) declarations. See the section on 'important' rules for more information.

3. Sort by origin: the author's style sheets override the user's style sheets which override the default style sheet. An imported style sheet has the same origin as the style sheet from which it is imported.

4. Sort by specificity of selector: more specific selectors will override more general ones. The definition and calculation of specificity is object-language dependent. Pseudo-elements and pseudo-classes are counted as normal elements and classes, respectively.

5. Sort by order specified: if two rules have the same weight, the latter specified wins. Rules in imported style sheets are considered to be before any rules in the style sheet itself.

The search for the property value must be terminated when any of the above steps yields a rule that has a higher weight than the other rules that apply to the same element/property combination.

This strategy gives author's style sheets considerably higher weight than those of the reader. It is therefore important that the User agent gives the user the ability to turn off the influence of a certain style sheet, e.g., through a pull-down menu.
Linking to external style sheet

HTML example:
```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <link href="http://web4health.info/web4health-v2.css" rel="stylesheet" type="text/css">
  <link href="http://web4health.info/site-map.css" rel="stylesheet" type="text/css">
</head>
</html>
```

XML example:
```xml
<?xml-stylesheet type="text/css" href="bach.css"?>
<ARTICLE>
  <HEADLINE>Fredrick the Great meets Bach</HEADLINE>
  <AUTHOR>Johann Nikolaus Forkel</AUTHOR>
  <PARA>
    ... ...
  </PARA>
</ARTICLE>
```

Shorthand properties in CSS

```css
h1 { font-weight: bold; font-size: 12pt; line-height: 14pt; font-family: Helvetica; font-variant: normal; font-style: normal; }
```

may be rewritten with a single shorthand property:
```css
h1 { font: bold 12pt/14pt Helvetica }
```

In this example, 'font-variant', and 'font-style' take their initial values.

HTML STYLE Attribute

Instead of
```html
<HEAD>
  <STYLE type="text/css">
    SPAN.xyz { font-size:10pt; font-style: bold }
  </STYLE>
</HEAD>
</BODY>
```

Here is one `<SPAN class="xyz">word</SPAN>` which is boldfaced and with a smaller font.

One can write
```html
Here is one `<SPAN style="font-size:10pt; font-style: bold">word</SPAN>` which is boldfaced and with a smaller font.
```

Disadvantage: Eliminates many of the advantages of separate style sheets.

Advantage: Easier for minor or automatic adoption of existing HTML code. Can use CSS commands in HTML.

Example of use: Google cached search results for the query “Olof Palme murder” using style attribute:

On February 28, 1986 Swedish Prime Minister Olof Palme was gunned down on a Stockholm street, as he was walking home from a cinema with his wife. While there are various theories about who could have been behind the murder, the identity of the culprit remains a mystery.

Special formatting of first line of a paragraph

CSS
```
p:first-line { text-transform: uppercase }
```

HTML
```
<p>Iamque fretum Minyae Pagasaea puppe secabant, perpetuaque trahens inopem sub nocte senectam Phineus visus erat, iuvenesque Aquilone creati virgineas volucres miseris ore fugarant.</p>
```

Rendering
```
IAMQUE FRETUM MINYAE PAGASAEA puppe secabant, perpetuaque trahens inopem sub nocte senectam Phineus visus erat, iuvenesque Aquilone creati virgineas volucres miseris ore fugarant.
```
**Special formatting of first letter of a paragraph**

**CSS**

```
p:first-letter {font-size: 3em }
```

**HTML**

```
<p>Iamque fretum Minyae Pagasaea puppe secabant, perpetuaque trahens inopem sub nocte senectam Phineus visus erat.</p>
```

Rendering

```
IAMQUE fretum Minyae Pagasaea puppe secabant, perpetuaque trahens inopem sub nocte senectam Phineus visus erat.
```

---

**Special formatting of first word of a paragraph**

**CSS**

```
span.largeword {font-size: 2em ;
text-transform: uppercase }
```

**HTML**

```
<p><span class=largeword>Iamque</span> fretum Minyae Pagasaea puppe secabant, perpetuaque trahens inopem sub nocte senectam Phineus visus erat.</p>
```

Rendering

```
IAMQUE fretum Minyae Pagasaea puppe secabant, perpetuaque trahens inopem sub nocte senectam Phineus visus erat.
```

---

**Different formatting for screen viewing and printing**

```
@media print {
.dummy {} /* some faulty browser need this */
.pagebreak
{ page-break-before: always }
.h1 { font-size: 18pt }
} /* end of media print */

@media screen, print {
.dummy {} 
.h1 {
 font-family: Verdana, Arial, Helvetica, sans-serif;
text-align: left;
padding-top: 0.2cm;
padding-bottom: 0.2cm;
font-weight: bold }

} /* end of media screen, print */

@media screen{
.dummy {}
.h1 {
 font-size: 26px }
} /* End of media screen */
```

---

**Box model**

```
Top
TM = Margin (Transparent)

TB = Border
TP = Padding

LM : LB : LP

Content

RP : RB : RM
```

---

-  Margin Edge
-  Border edge
-  Padding edge
-  Content edge
Example of use of box model

```
<style type="text/css">
div.mybox {
    margin: 1cm 3cm; border: 5pt solid red;
    padding: 5mm; background-color: #CCCCCC;
    text-align: center; font-size: 20pt }
</style>
</head><body>

Iamque fretum Minyae Pagasaea puppe secabant.</body>
```

## The display property

- **block**: A block box (laid out vertically).
- **inline**: One or more inline boxes (laid out horizontally). Split into several boxes if all does not fit into the line.
- **inline-block**: A block box, which itself is flowed as a single inline box, similar to a replaced element. The inside of an inline-block is formatted as a block box, and the element itself is formatted as a replaced element on the line.
- **list-item**: A list of items
- **run-in**: Either block or in-line depending on context, example: TO BE OR NOT TO BE, that is the question.
- **table**: As HTML `<TABLE>`
- **table-row**: As HTML `<TR>`
- **table-cell**: As HTML `<TD>`

### Inline and block boxes

```
<style type="text/css">
.inlinebox { display: Inline; padding: 5mm }
.blockbox { display: block; border: 6pt solid red }
</style></head><body>

Iamque fretum Minyae Pagasaea puppe secabant.
```

### Static and relative positioning

#### position:
- **static**: Normal flow
- **relative**: Normal flow plus offset
- **absolute**: Fixed position in relation to surrounding block
- **fixed**: Fixed position in relation to:
  - Window (screen)
  - Page (printing)

Combined with top, bottom, right, left which can have:
- absolute value
- percentage (of containing block)
- auto
- inherit
Side by side effects:

Side by side: Why does absolute positioning not work?

```html
<div style="margin:auto; width: 200px">
  <div style="position:absolute; left: 0px; width:50px">
    <p>Text in column one.</p>
  </div>
  <div style="position:absolute; left: 55px; width:50px">
    <p>Text in column two.</p>
  </div>
  <div style="position:absolute; left: 110px; width:50px">
    <p>Text in column three.</p>
  </div>
</div>

Note:
"margin-left" works somewhat better than "left", why?

Side by side: Table cells work with most browsers except Explorer

```html
<html><head>
<style type="text/css">
.table { display:table; margin: auto; }
.row { display:table-row; }
.cell { display:table-cell; width:50px; padding:5px; }
#footer { text-align:center; }
</style>
</head>
<body>
<div class="table">
  <div class="row">
    <div class="cell">Text in column one.</div>
    <div class="cell">Text in column two.</div>
    <div class="cell">Text in column three.</div>
  </div>
</div>
<div id="footer">
  <p>Text below the table</p>
</div>
</body></html>
```

Float

```html
<STYLE type="text/css">
IMG { float: left } BODY, P, IMG { margin: 1em }
</STYLE></HEAD></BODY>
<p>
<IMG src="gunborg-palme-154px.jpg" alt="A floating picture">
Iamque fretum Minyae Pagasae puppe secabant, perpetuque
trahens inopem sub nocte senectam Phineus visus erat,
iuvenescque Aquilone creati virgineas volucres miseri sensis ore
fugarant, multaque perpessi ...
```

Side by side: Table cells work with most browsers except Explorer

```html
<html><head>
<style type="text/css">
.table { display:table; margin: auto; }
.row { display:table-row; }
.cell { display:table-cell; width:50px; padding:5px; }
#footer { text-align:center; }
</style>
</head>
<body>
<div class="table">
  <div class="row">
    <div class="cell">Text in column one.</div>
    <div class="cell">Text in column two.</div>
    <div class="cell">Text in column three.</div>
  </div>
</div>
<div id="footer">
  <p>Text below the table</p>
</div>
</body></html>
```
Layering

```
<HTML><HEAD><STYLE type="text/css">
  .stackone { position: absolute; left: 10px; top: 10px; width: 120px; height: 128px }
  .stacktwo { position: absolute; left: 140px; top: 10px; width: 120px; height: 128px }
</STYLE></HEAD><BODY>
  <IMG src="eatfem.gif" alt="Eating" class=stackone style="z-index: 2">
  <IMG src="flag.gif" style="vertical-align: middle">text-top: <IMG src="flag.gif" style="vertical-align: text-top">
  <DIV class=stacktwo style="z-index: 1">
    eat eat eat eat eat eat eat eat eat eat eat eat eat eat eat
  <DIV class=stacktwo style="z-index: 2">
  <IMG src="eatfem.gif" alt="Eating" class=stackone style="z-index: 2">
</BODY></HTML>
```

vertical-align

```
<IMG src="EATFEM.GIF" style="vertical-align:middle">
Middle: <IMG src="flag.gif" style="vertical-align: middle">
baseline: <IMG src="flag.gif" style="vertical-align: baseline">
text-top: <IMG src="flag.gif" style="vertical-align: text-top">
text-bottom: <IMG src="flag.gif" style="vertical-align: text-bottom">
top: <IMG src="flag.gif" style="vertical-align: top">
```

text-indent

```
<STYLE type="text/css">
  body { width: 50ex }
  p { line-height: 130%; text-indent: 2em; padding: 0px; margin: 0px }
</STYLE>

<p>Iamque fretum Minyae Pagasaea puppe secabant .....</p>
<p>Multaque perpessi claro sub Iasone tandem .....</p>
```

```
Iamque fretum Minyae Pagasaea puppe secabant,
perpetuaque trahens inopem sub nocte senectam
Phineus visus eat.
Iuvenesque Aquilone creati virgineas voluptus
miseri senis ore fugarent.
Multaque perpessi claro sub Iasone tandem
contigerant rapidas limosi Phasidos undas.
Dumque adeunt regem Phrixeaque vellera poscunt ....</p>
```

By text-top baseline text-bottom

Border styles

```
<STYLE type="text/css">
  body { margin: 20px; font-family: sans-serif; font-size: 24px }
</STYLE>

<p>Dumque adeunt regem Phrixeaque vellera poscunt ....</p>
```

```
<SPAN style="border:solid 10px">solid</SPAN>
<SPAN style="border:dotted 10px">dotted</SPAN>
<SPAN style="border:dashed 5px">dashed</SPAN>
<SPAN style="border:double 10px">double</SPAN>
<SPAN style="border:groove 10px">groove</SPAN>
<SPAN style="border:ridge 10px">ridge</SPAN>
```